The HLA R&D Study and Prototype

Michael Reid

Computer Sciences Corporation

Mike.Reid@gsfc.nasa.gov

September 29, 2000

What is the HLA?

- The High Level Architecture (HLA) is a DoD and IEEE standard framework that supports modeling and simulation.
- "The HLA is the glue that allows you to combine computer simulations into a larger simulation."[1]

1. F. Kuhl, R. Weatherly, J. Dahmann, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, p. 1, Prentice Hall PTR, 1999.

What is the HLA (continued)?

- A standard architecture and framework (IEEE-1516)
- An integration system
- A communications and data sharing protocol
- A synchronization and time management system
- A distributed computing environment

Role of the HLA

- Provides a standard architecture for use in large-scale simulation systems
- Facilitates distributed and multi-platform computing in simulation systems
- Supports collaborative development
- Integrates separate and remote applications
- Development sponsored by the DoD's Defense Modeling and Simulations Office (DMSO)

The HLA and NASA

- Integrating numerous specialized simulators into larger overall space mission simulations
- Modeling natural Earth or planetary geophysical systems
- Standard for integrating simulators of various spacecraft and instruments
- Constellation and formation flying simulations

HLA Terminology

- Federate
  - An individual simulator application.
- Federation
  - A simulation composed of two or more (often many more) federates integrated together.
- Federation Execution
  - A session in which a federation is running.

HLA Terminology (continued)

- Federation Object Model (FOM)
  - The common object model which defines the data shared by federates within the federation
- Simulation Object Model (SOM)
  - The object model which defines the data shared by an individual federate with the federation
- Runtime Infrastructure (RTI)
  - The COTS or GOTS software which runs an HLA-compliant simulation

The R&D Project

- Studied the HLA
- Developed an HLA-compliant prototype space mission simulator
- Evaluated the HLA's strengths and weaknesses
- Evaluated its applicability to NASA's simulation needs
- Evaluated Linux™ as a development environment and the use of free, open-source software

The R&D Prototype

- Developed a common HLA-interface module
- Created four fully HLA-compliant simulation applications (federates)
  - Spacecraft Controller
  - Orbit Calculator
  - Tracking Station
  - Earth
- Made the four federates “play” together in a federation
- A clock-driven simulation
- Uses the DMSO RTI

The Spacecraft Controller

- Creates two spacecraft HLA Objects
  - "Landsat 7"
  - "Terra"
- Each spacecraft object contains attributes
  - Current geographic positional data
  - An onboard "magnetometer instrument"
- Displays a simple graphic of the Earth magnetic field intensity as measured at a given moment as measured by the "magnetometer instrument"

The Orbit Calculator

- Reads a file containing the precomputed geographic positions of the two spacecraft at a given moment
- The positional data is generated using Matlab® programs obtained from the Flight Dynamics group using orbital elements for Landsat 7 and Terra
- Updates the positional data of both “spacecraft” at each increment of the simulation time
- Paces the simulation based on the orbital positions and times

The Tracking Station

- Reads the positional attributes of the spacecraft objects
- Displays a map of the Earth and plots the current positions and tracks of the two satellites as time advances
- Paced by, but does not pace the federation

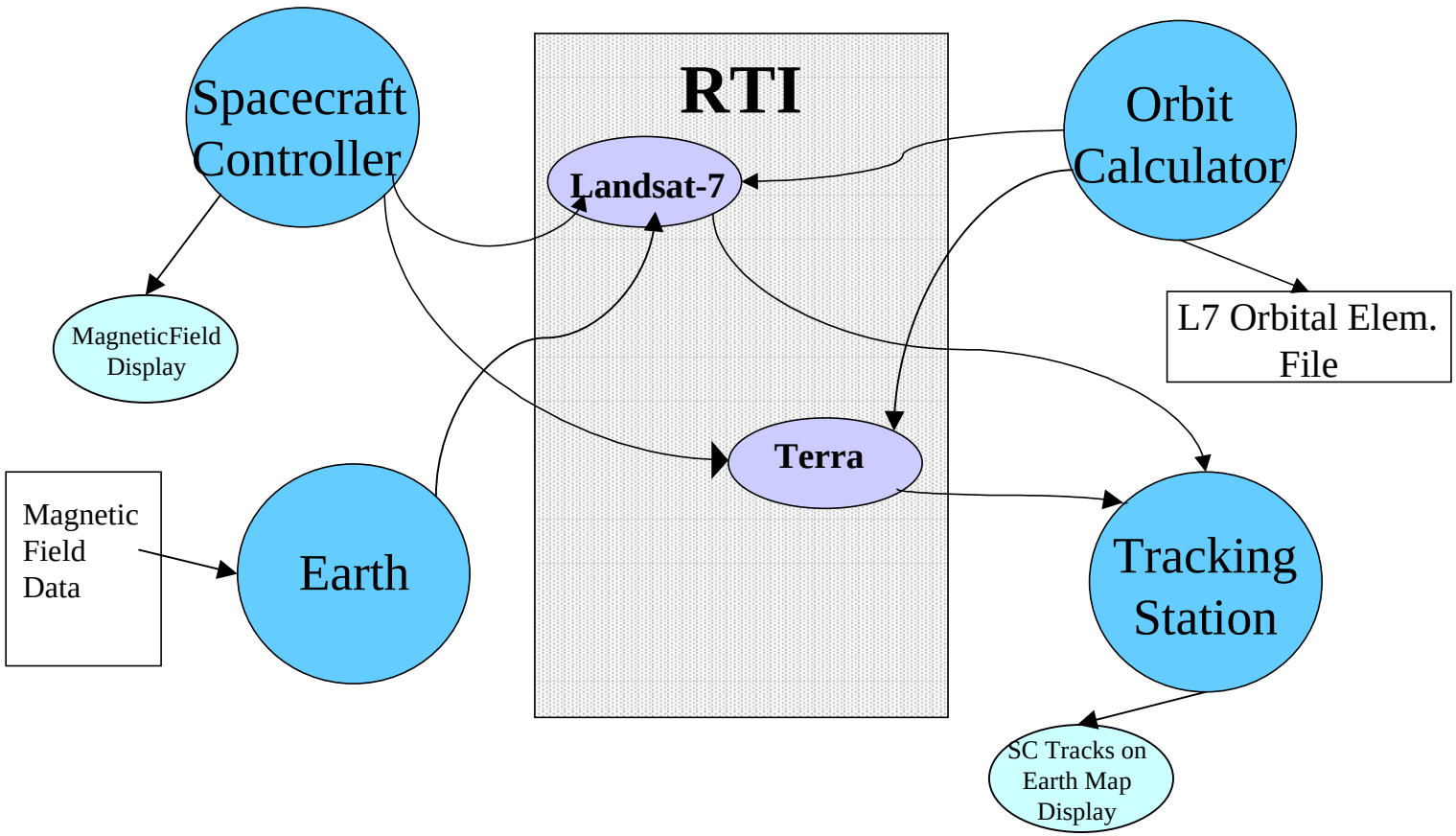
The Earth

- Simulates the incorporation of Earth science data
- Generates magnetic field intensity data
  - Reads files containing precomputed magnetic field data
  - Magnetic field data created by Matlab® programs obtained from the Flight Dynamics group
- Updates the “magnetometer instrument” in the spacecraft objects
  - Simulates the instrument collecting science data

The Spacecraft Objects

- HLA Objects defined in the FOM
- Very loosely patterned after Landsat 7 and Terra
- Geographic positional data attributes
  - Created by the Spacecraft Controller, updated by the Orbit Calculator, read by the Tracking Station
  - Displayed on a map of the Earth as tracks
- Magnetometer Instrument
  - Simulates the collection of science data
  - Created and read by the Spacecraft Controller, updated by the Earth federate
  - Displayed in a line chart by the Spacecraft Controller

Diagram of the Federation



Software Used in the R&D Prototype

- Prototype built using entirely FREE software!
  - Linux™ (GPL)
    - Unix-like, POSIX-compliant operating system
  - GCC C++ compiler (GPL)
    - Excellent free C++ compiler
  - Octave (GPL)
    - 4GL for math and science, mostly source compatible with Matlab®
    - Used for running the Matlab®-based Flight Dynamics programs
  - gnuplot (other freeware license)
    - Free graphics program used for the graphics
  - DMSO RTI (GOTS)
    - Fully implemented and tested HLA-compliant RTI

GPL: *GNU General Public License* (open source license, allows free use, copying, and distribution)

Accomplishments of the HLA Study and Prototyping Project

- Learned how to build HLA-compliant applications
- Developed a fully reusable interface module for quickly constructing HLA federates
- Discovered the HLA's strengths and weaknesses
- Gained an understanding of the types of applications for which the HLA is an appropriate technology
- Came up with ideas on how to make the SIMSS HLA-compliant
- Evaluated Linux and the free, open-source software paradigm

What the HLA Is

- A standard architecture and framework (IEEE-1516)
- An integration system
- A communications and data sharing protocol
- A synchronization and time management system
- A distributed computing environment

What the HLA Is Not

- A simulator
- A modeling tool
- A telemetry generator
- A data display and analysis tool
- A real-time integration system
- A “plug and play” system
- A user interface

What the HLA is Good For

- Game-type simulations
  - A little “universe” is being simulated
  - Numerous actors are coming and going, interacting, and sharing data
- Integrating multiple simulators together into a larger, distributed simulation system
- Scalability

What the HLA is Not Good For

- Simulations which are not game-like
- Simulations not composed of multiple, separate, and interacting parts
- Applications requiring specific interfaces to external components (e.g., TCP, serial, etc.)
- Applications requiring high-performance or near real-time throughput

What the HLA Will Do

- Handle the communications and data sharing between remote and disparate applications via a standard interface
- Manage the ownership, updates, distribution, and persistence of shared data
- Provide time management and synchronization
- Make it more practical to reuse preexisting simulators provided that they are HLA-compliant

What the HLA Will Not Do

- Build your simulation for you
- Increase the fidelity of your simulators
- Generate your data
- Display your data
- Provide a user interface
- Make “plug and play” a reality

Pluses

- Complete infrastructure for running distributed simulations
- A DoD and industry (IEEE-1516) standard
- Runs on multiple platforms
- Good technical support from DMSO
- Handles all of the communications and data management between applications

Minuses

- Current RTI is too slow and not stable enough for real-time or mission-critical applications
- RTI is very sensitive to the system's network configuration
- Best used on a closed LAN
- Poor error messages
- Consumes a lot of system resources
- Steep learning curve
- Requires a significant amount of programming
- Sparse documentation

The HLA and NASA

- Integrating numerous specialized simulators into larger overall space mission simulations
  - COTS Tools
  - Telemetry generation (e.g., SIMSS)
  - Instrument simulators
  - Science data generators
  - Ground system simulators
- Modeling natural Earth or planetary geophysical systems
  - Weather systems and climatic factors
  - Geologic activity
  - Biological factors
  - Cosmic influences

Formation Flying

- Involve numerous spacecraft interacting with each other
- Numerous ground stations interacting
- Natural natural objects
- The space environment
- HLA could integrate numerous simulators to model overall missions

An Envisioned Simulation

- Numerous applications manage multiple spacecraft and ground station objects
- Other simulators model the natural objects and the space environment
  - Earth, Sun, Moon, planets, astronomical objects
  - Radiation, solar wind, gravitational and magnetic fields, interplanetary dust, etc.

A Larger NASA Simulation

- COTS products are integrated
- Various simulators run remotely in a distributed system and on multiple platforms
  - Unix, Linux, WinNT
- Multiple and remote organizations collaborate
- RTI handles all networking

The HLA and the SIMSS

- How to make the SIMSS HLA-compliant
  - Create a separate interface process, which communicates with the SIMSS using IP sockets on one side and with the HLA on the other
    - OR … Add an HLA interface to the SIMSS as an alternative to IP sockets
  - Create a SOM for the SIMSS
- One or more SIMSS spacecraft components could run as federates as part of a larger simulation (federation)
- An HLA-compliant SIMSS component could generate telemetry for a spacecraft object while other applications, including COTS, update the object's other attributes

Conclusions on the HLA

- The HLA is a viable technology for certain types of simulations and modeling
  - Game-type simulations
  - Modeling a “universe” where numerous actors interact with each other and come and go
  - Distributed systems
  - Systems involving different computing platforms
- The HLA is not well suited for other applications
  - Single component or single task simulators
  - Real-time systems
  - Integrating hardware components

Conclusions on Linux and the Open-source Model

- Linux is an excellent platform for development and operations
  - Nearly perfect clone of Unix and it's FREE!!!
  - Very stable and well documented
  - Uses system resources efficiently
  - Lots of high-quality, free, and open-source software for it
  - The GNU GCC C++ compiler is one of the best
  - Plenty of free software development tools from GNU and others
- Using Linux and GNU software is low-risk and provides tremendous cost savings
- Easily scales up to higher-end Unix-based systems

Further Information

- HLA
  - M. Reid, *An evaluation of the high level architecture (HLA) as a technology for use in space mission simulations: results and conclusions from the SIMSS HLA study and prototyping effort*, (August 2000). White paper available at: http://cmex.gsfc.nasa.gov/TECHHIGH/HLA/HLA_Conclusions.pdf
  - DMSO HLA Website: http://hla.dmso.mil
  - F. Kuhl, R. Weatherly, J. Dahmann, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, Prentice Hall PTR, Upper Saddle River, NJ (1999)
- Open Source Software
  - Free Software Foundation: http://www.gnu.org
  - Linux: http://www.li.org